



**UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO**

DIPARTIMENTO DI INFORMATICA

CORSO DI LAUREA IN INFORMATICA

TESI DI LAUREA
IN
SISTEMI AD AGENTI

**Sviluppo e Gestione di Chatbot
Addestrati con Tecnica RAGG
tramite Servizi RESTful**

RELATORI:

Prof.ssa Berardina Nadja DE CAROLIS

LAUREANDO:

Federico Calò

ANNO ACCADEMICO 2023 - 2024

Indice

Introduzione	1
1 Introduzione ai LLMs e ai servizi REST	3
1.1 Storia e caratteristiche degli LLMs	4
1.1.1 Applicazioni degli LLM	5
1.2 Storia e caratteristiche dei servizi REST	7
1.2.1 Applicazioni dei servizi REST	8
1.3 Servizi REST e LLMs	10
2 Analisi e progettazione	13
2.1 Scelta tecnologica	14
2.1.1 Creazione e addestramento dei chatbot	15
3 Titolo capitolo 3	17
3.1 Titolo Paragrafo	17
3.1.1 Titolo Sottoparagrafo	18
4 Titolo capitolo 4	19
4.1 Titolo Paragrafo	19
4.1.1 Titolo Sottoparagrafo	20
5 Conclusioni	21

Introduzione

Nel contesto odierno, caratterizzato da una crescente interazione tra utenti e sistemi digitali, i chatbot stanno emergendo come strumenti fondamentali per migliorare l'esperienza dell'utente e ottimizzare i processi aziendali. Tuttavia, la creazione e gestione di chatbot avanzati richiede un approccio integrato che sfrutti le migliori tecniche di addestramento e architetture software.

Questa tesi si propone di affrontare tale sfida attraverso la progettazione e realizzazione di un ecosistema avanzato per la creazione e gestione di chatbot. L'obiettivo principale del progetto è sviluppare una piattaforma che consenta la creazione di diversi chatbot addestrati utilizzando la metodologia Retrieval-Augmented Generation (RAG). Questa tecnica, che combina la capacità di generazione di testo con la capacità di recupero di informazioni, permette di creare chatbot più intelligenti e contestualmente pertinenti.

Per garantire una fruizione ottimale e una gestione efficiente dei chatbot, è stata progettata una soluzione integrata che comprende:

- **Frontend in ReactJS:** Un'interfaccia utente interattiva e dinamica che facilita l'interazione degli utenti con i chatbot. ReactJS è stato scelto per la sua capacità di creare applicazioni web veloci e reattive.
- **Servizi REST API:** Interfacce di programmazione che permettono la comunicazione tra il frontend e i servizi backend. Le API sono state sviluppate utilizzando due tecnologie complementari: Spring Boot e Python. Spring Boot è stato utilizzato per costruire servizi backend robusti e scalabili, mentre Python è stato impiegato per implementare logiche specifiche e processi di machine learning legati ai chatbot.

- **Metodologia RAG:** L'approccio Retrieval-Augmented Generation è stato adottato per addestrare i chatbot, combinando tecniche di recupero di informazioni e generazione di risposte, al fine di migliorare la pertinenza e l'accuratezza delle risposte fornite dai bot.

La realizzazione di questo ecosistema richiede un'integrazione armoniosa di componenti software e tecniche di machine learning, mirata a offrire una piattaforma versatile e potente per lo sviluppo di chatbot avanzati. L'importanza di questo progetto risiede nella sua capacità di semplificare la creazione e gestione di chatbot sofisticati, migliorando significativamente l'interazione uomo-macchina e offrendo soluzioni scalabili per diverse applicazioni aziendali e di consumo.

Capitolo 1

Intrroduzione ai LLMs e ai servizi REST

I Large Language Models (LLMs) rappresentano una classe avanzata di modelli di intelligenza artificiale progettati per comprendere, generare e interagire con il linguaggio umano a livelli di complessità senza precedenti. Questi modelli sono stati all'avanguardia delle ricerche nel campo del Natural Language Processing (NLP) e sono stati progettati per comprendere, interpretare, generare e tradurre testi in linguaggio naturale. La loro "grandezza" (large) non deriva solo dalla dimensione del modello in termini di numero di parametri, ma anche dall'enorme quantità di dati su cui vengono addestrati. Questi modelli apprendono autonomamente le strutture linguistiche e le relazioni semantiche dai dati, permettendo loro di generare testi coerenti e pertinenti, rispondere a domande, riassumere documenti e molto altro.

I servizi RESTful (Representational State Transfer) rappresentano un'architettura per la creazione di applicazioni web e API che consente una comunicazione semplice ed efficiente tra client e server. Basati sui principi dell'architettura REST, questi servizi utilizzano il protocollo HTTP per gestire le richieste e le risposte tra le diverse componenti di un sistema, fornendo quindi un modo standardizzato e scalabile per la comunicazione tra queste parti. Essi giocano un ruolo cruciale nella connessione tra il frontend e il backend di sistemi complessi, come quelli che utilizzano modelli di linguaggio avanzati. Per esempio, nel contesto di un chatbot addestrato con la metodologia RAG, i servizi RESTful possono facilitare la comunicazione tra l'interfaccia utente (frontend) e i servizi di elaborazione e generazione

del linguaggio (backend), garantendo un'interazione fluida e reattiva.

1.1 Storia e caratteristiche degli LLMs

Il viaggio dei Large Language Models (LLMs) inizia negli anni '50, quando l'intelligenza artificiale era ancora agli albori. In quei primi giorni, i ricercatori si concentravano principalmente su sistemi basati su regole e approcci simbolici. I modelli di linguaggio dell'epoca erano piuttosto rudimentali e si basavano su semplici regole predefinite, senza la capacità di apprendere direttamente dai dati.

Con l'avvento del machine learning negli anni '80 e '90, le cose iniziarono a cambiare. I modelli statistici, come i modelli di Markov nascosti e gli n-grammi, iniziarono a essere utilizzati per l'elaborazione del linguaggio naturale (NLP). Sebbene questi modelli offrissero un miglioramento rispetto ai sistemi basati su regole, erano ancora limitati nella loro capacità di comprendere e generare linguaggio naturale in modo complesso.

La vera rivoluzione arrivò con l'introduzione delle reti neurali e del deep learning nei primi anni 2000. Architetture come le reti neurali ricorrenti (RNN) e le LSTM (Long Short-Term Memory) iniziarono a gestire meglio le sequenze di testo e le dipendenze a lungo termine, portando a progressi significativi nella NLP.

Ma è stato il 2017 a segnare una svolta decisiva con l'introduzione dell'architettura dei trasformatori nel paper "Attention is All You Need" di Vaswani et al. Questa innovazione ha rivoluzionato il campo, aprendo la strada a modelli come BERT (Bidirectional Encoder Representations from Transformers) e GPT (Generative Pre-trained Transformer). Questi modelli hanno dimostrato capacità straordinarie nella gestione del linguaggio, grazie alla loro abilità di catturare contesti complessi e gestire grandi quantità di dati.

Negli ultimi anni, i LLMs hanno raggiunto nuove vette con modelli di enormi dimensioni come GPT-3 e GPT-4. Questi modelli, con i loro miliardi di parametri, sono stati addestrati su enormi corpus di dati, permettendo loro di comprendere e generare testo con una precisione e coerenza senza precedenti.

Le caratteristiche distintive dei LLMs sono davvero notevoli. Innanzitutto, la loro grandezza è impressionante: modelli come GPT-3 hanno 175 miliardi di parametri, una quantità enorme che consente di catturare e rappresentare una vasta

gamma di conoscenze linguistiche e semantiche. Questo grande numero di parametri è fondamentale per permettere ai modelli di comprendere e generare testo in modo accurato e fluido.

Inoltre, gli LLMs sono addestrati su enormi dataset provenienti da una varietà di fonti, tra cui libri, articoli e conversazioni online. Questo ampio addestramento consente ai modelli di apprendere un'ampia gamma di stili e contenuti linguistici.

Una delle caratteristiche più avanzate è la loro capacità di comprensione e generazione del linguaggio. Gli LLMs non solo comprendono il contesto delle conversazioni e dei testi, ma possono anche generare risposte pertinenti e coerenti, eseguire traduzioni, riassumere documenti e persino creare testi originali.

I LLMs utilizzano tecniche di apprendimento non supervisionato e pre-addestramento per costruire una base solida di conoscenze linguistiche. Questo approccio consente ai modelli di acquisire una comprensione generale del linguaggio prima di essere affinati su compiti specifici, migliorando ulteriormente la loro capacità di applicarsi a vari compiti di NLP.

Infine, l'architettura dei trasformatori, alla base della maggior parte degli LLMs moderni, è fondamentale. I trasformatori utilizzano meccanismi di attenzione per pesare l'importanza delle diverse parti del testo, gestendo in modo più efficace le dipendenze a lungo termine e migliorando la comprensione del contesto.

In sintesi, i Large Language Models hanno rappresentato una delle più grandi innovazioni nell'ambito dell'intelligenza artificiale. La loro evoluzione ha trasformato il modo in cui interagiamo con il linguaggio e ha aperto nuove possibilità per applicazioni in vari ambiti, dall'assistenza clienti alla generazione di contenuti creativi.

1.1.1 Applicazioni degli LLM

I Large Language Models (LLMs) hanno trasformato radicalmente il campo dell'intelligenza artificiale e dell'elaborazione del linguaggio naturale (NLP), aprendo nuove possibilità applicative in vari settori. Questi modelli, grazie alla loro capacità di comprendere e generare testo con una precisione senza precedenti, sono diventati strumenti essenziali per una vasta gamma di applicazioni.

Una delle principali applicazioni degli LLMs è nei sistemi di assistenza virtuale. Tecnologie come Siri, Alexa e Google Assistant utilizzano LLMs per comprendere e

rispondere alle richieste degli utenti, eseguire comandi vocali, fornire informazioni in tempo reale e interagire con altre applicazioni. Questi assistenti virtuali si basano sulla capacità dei LLMs di interpretare il linguaggio naturale, comprendere il contesto e generare risposte appropriate in modo fluido e naturale.

Nel settore della generazione di contenuti, i LLMs vengono utilizzati per creare testi automaticamente. Ad esempio, possono generare articoli di notizie, post sui social media, descrizioni di prodotti, e persino testi creativi come poesie o storie. Strumenti come GPT-3 hanno dimostrato di poter produrre contenuti di alta qualità che sono quasi indistinguibili da quelli scritti da esseri umani, accelerando il processo di creazione di contenuti e rendendolo più accessibile.

Gli LLMs sono anche ampiamente utilizzati nei sistemi di traduzione automatica. Modelli come quelli implementati da Google Translate e altri servizi di traduzione online sono in grado di tradurre testi da una lingua all'altra con una precisione e fluidità sempre maggiori. Questi modelli sfruttano la loro comprensione approfondita delle strutture linguistiche e delle relazioni semantiche per produrre traduzioni che rispettano il contesto e il significato originale.

Nel campo della analisi del sentiment e del monitoraggio dei social media, i LLMs vengono utilizzati per analizzare grandi volumi di dati testuali, come post sui social media, recensioni dei clienti e articoli di opinione, per comprendere l'opinione pubblica e il sentiment generale riguardo a prodotti, marchi o eventi. Questa analisi permette alle aziende di monitorare la percezione del pubblico e di prendere decisioni informate basate sui feedback raccolti.

Un'altra applicazione emergente degli LLMs è nel supporto alla scrittura e alla ricerca. Strumenti come quelli sviluppati da OpenAI vengono utilizzati per assistere scrittori, ricercatori e studenti nel creare bozze, riassumere articoli, generare idee o formulare risposte a domande complesse. Questi modelli possono accelerare il processo creativo e migliorare la produttività, fornendo suggerimenti pertinenti e contenuti coerenti in tempo reale.

Infine, gli LLMs trovano applicazione anche nel supporto decisionale aziendale. Attraverso l'analisi di dati testuali e documenti, questi modelli possono aiutare le aziende a estrarre informazioni critiche, fare previsioni basate su trend emergenti e supportare la presa di decisioni strategiche. Questo è particolarmente utile in contesti come l'analisi di mercato, la gestione del rischio e la pianificazione

aziendale.

1.2 Storia e caratteristiche dei servizi REST

Nel panorama attuale dello sviluppo software, i servizi RESTful hanno acquisito un ruolo fondamentale grazie alla loro semplicità ed efficacia nel facilitare la comunicazione tra diverse applicazioni e componenti di un sistema. L'architettura REST, acronimo di Representational State Transfer, è stata introdotta da Roy Fielding nel 2000 e ha rapidamente rivoluzionato il modo in cui le applicazioni web sono progettate e sviluppate.

Un aspetto chiave dei servizi RESTful è la loro natura stateless. Questo significa che ogni richiesta effettuata dal client verso il server è indipendente dalle altre; il server non conserva informazioni sullo stato delle richieste precedenti. Questa caratteristica semplifica notevolmente la scalabilità delle applicazioni, poiché non è necessario che il server gestisca sessioni o mantenga traccia delle interazioni passate. Ogni richiesta contiene tutte le informazioni necessarie per essere processata in modo autonomo.

La separazione tra client e server è un altro principio cardine di REST. In questa architettura, il client è responsabile della presentazione e dell'interfaccia utente, mentre il server gestisce la logica di business e l'archiviazione dei dati. Questo tipo di suddivisione permette uno sviluppo più modulare, in cui le componenti del frontend e del backend possono evolvere in modo indipendente. Ad esempio, è possibile aggiornare l'interfaccia utente senza dover necessariamente intervenire sulla logica del server, e viceversa.

Un'altra caratteristica distintiva dei servizi RESTful è l'utilizzo di un'interfaccia uniforme per la comunicazione. I metodi standard del protocollo HTTP, come GET, POST, PUT e DELETE, sono utilizzati per eseguire operazioni su risorse identificate da URL specifici. Questo approccio non solo semplifica lo sviluppo, ma rende anche le API RESTful facili da comprendere e utilizzare, poiché seguono convenzioni ben definite e universalmente riconosciute.

Le risorse, in un'architettura RESTful, sono rappresentate tramite URL. Ogni risorsa, che può essere un oggetto, un servizio, o una qualsiasi entità gestita dal

server, è accessibile tramite un URL univoco. Questo rende l'accesso e la manipolazione delle risorse intuitivi e facilmente gestibili, poiché tutto è strutturato attorno a un sistema di indirizzamento chiaro e organizzato.

Un ulteriore vantaggio offerto dai servizi RESTful è la possibilità di caching. Le risposte alle richieste possono essere memorizzate dal client per un certo periodo di tempo, riducendo così la necessità di ripetere richieste identiche e migliorando l'efficienza complessiva del sistema. Il caching permette di alleviare il carico sui server e di velocizzare l'accesso ai dati, migliorando l'esperienza utente.

Infine, l'architettura REST supporta la realizzazione di sistemi stratificati. Ciò significa che tra il client e il server possono essere introdotti vari componenti intermediari, come proxy e gateway, che svolgono funzioni aggiuntive come la gestione della sicurezza, il bilanciamento del carico e l'ottimizzazione delle prestazioni. Questa stratificazione permette di scalare il sistema in modo più efficiente e di aggiungere nuove funzionalità senza interrompere l'intera infrastruttura.

Nel complesso, i servizi RESTful hanno rivoluzionato il modo in cui le applicazioni web e mobile interagiscono con il backend. La loro semplicità, combinata con una grande flessibilità, li rende ideali per la costruzione di sistemi complessi e distribuiti, come i microservizi, che possono essere sviluppati, testati e aggiornati indipendentemente. La diffusione di questa architettura ha consentito lo sviluppo di applicazioni altamente scalabili e manutenibili, in grado di adattarsi rapidamente alle esigenze in continua evoluzione del mondo digitale.

1.2.1 Applicazioni dei servizi REST

I servizi RESTful hanno trovato un'ampia applicazione in numerosi settori e tipologie di sistemi, grazie alla loro flessibilità, semplicità e capacità di facilitare l'integrazione tra diverse componenti software. L'approccio REST è diventato lo standard de facto per la progettazione di API (Application Programming Interface), utilizzate per collegare diverse applicazioni, servizi e dispositivi in modo efficiente e scalabile.

Una delle applicazioni più comuni dei servizi REST è nello sviluppo di applicazioni web e mobile. Le API RESTful consentono a queste applicazioni di comunicare con il backend per recuperare dati, inviare informazioni e interagire con altri servizi. Ad esempio, un'app mobile di social media utilizza REST per caricare i feed degli

utenti, pubblicare nuovi post, gestire notifiche e interagire con altre funzionalità della piattaforma. La natura stateless di REST rende possibile gestire un elevato numero di richieste simultanee, rendendo le applicazioni più rapide e reattive.

I servizi RESTful sono anche ampiamente utilizzati nei microservizi, un'architettura che scompone le applicazioni in una serie di piccoli servizi indipendenti, ciascuno dei quali svolge una funzione specifica. Questa struttura permette di sviluppare, distribuire e scalare i componenti in modo indipendente, riducendo la complessità e migliorando l'agilità del sistema. In questo contesto, REST funge da collante che permette ai vari microservizi di comunicare tra loro in modo standardizzato e coerente.

Nel campo dell'Internet of Things (IoT), REST viene spesso impiegato per facilitare la comunicazione tra dispositivi intelligenti e i sistemi backend che li gestiscono. Ad esempio, i dispositivi IoT, come termostati intelligenti o sistemi di monitoraggio remoto, utilizzano API RESTful per inviare dati sensoriali a un server centrale, che li elabora e restituisce comandi o configurazioni aggiornate. La leggerezza e la scalabilità di REST lo rendono ideale per gestire reti di dispositivi distribuiti e a bassa potenza.

Anche nel settore delle integrazioni tra sistemi eterogenei, i servizi RESTful sono ampiamente utilizzati. Grazie alla loro natura universale e indipendente dalla piattaforma, REST permette a sistemi sviluppati con tecnologie diverse di comunicare tra loro. Ad esempio, un sistema di gestione delle risorse umane potrebbe utilizzare REST per sincronizzare dati con un sistema di contabilità o con applicazioni di terze parti per la gestione dei salari.

Infine, REST è largamente impiegato nella creazione di applicazioni cloud. I servizi cloud come AWS, Google Cloud e Microsoft Azure offrono API RESTful per gestire risorse cloud, come server, database e servizi di storage. Queste API permettono agli sviluppatori di automatizzare la gestione delle risorse cloud, implementando funzioni di scalabilità automatica, monitoraggio delle performance e gestione delle configurazioni.

1.3 Servizi REST e LLMs

L'interazione tra servizi RESTful e Large Language Models (LLMs) rappresenta un approccio potente e versatile per sfruttare le capacità avanzate dell'intelligenza artificiale in applicazioni web e mobile. I servizi REST, grazie alla loro natura semplice e standardizzata, forniscono un'infrastruttura ideale per orchestrare e gestire la comunicazione tra i client e i modelli di intelligenza artificiale, come gli LLMs, che sono in grado di comprendere e generare linguaggio naturale.

In un'architettura in cui i servizi RESTful fungono da intermediari tra l'interfaccia utente e i LLMs, ogni richiesta effettuata dal client può essere facilmente instradata verso il modello AI appropriato per l'elaborazione. Ad esempio, un utente potrebbe inviare una domanda o un comando tramite un'applicazione web o mobile. Questa richiesta viene trasmessa tramite una chiamata API RESTful al server, dove i servizi REST gestiscono la richiesta e la inoltrano all'LLM per l'elaborazione.

Uno dei principali vantaggi dell'utilizzo di servizi REST in combinazione con LLMs è la loro capacità di gestire richieste stateless. Questo significa che ogni richiesta inviata al server è indipendente e contiene tutte le informazioni necessarie per essere elaborata. Gli LLMs, che possono richiedere risorse computazionali significative, possono così concentrarsi sull'elaborazione del linguaggio naturale senza doversi preoccupare dello stato della sessione utente. Questo approccio semplifica l'infrastruttura e migliora la scalabilità del sistema.

Un'altra area in cui i servizi REST possono migliorare l'integrazione con gli LLMs è attraverso la modularità. I servizi RESTful possono essere progettati per svolgere funzioni specifiche, come l'autenticazione, la gestione delle richieste, o l'elaborazione di dati prima o dopo l'intervento degli LLMs. Questa modularità consente di creare pipeline di elaborazione flessibili, in cui diverse componenti possono essere aggiornate o sostituite senza interrompere l'intero sistema. Per esempio, è possibile implementare un servizio REST che pre-processa i dati dell'utente, rendendoli più facilmente comprensibili per l'LLM, migliorando così la precisione delle risposte generate.

Inoltre, i servizi RESTful possono facilitare l'integrazione di LLMs con altre risorse e servizi. Ad esempio, un'API REST può essere utilizzata per recuperare dati da un database o da una fonte esterna, che poi vengono combinati con le

capacità generative dell'LLM per fornire risposte più complete e contestuali. Questo approccio, noto come Retrieval-Augmented Generation (RAG), permette agli LLMs di accedere a informazioni specifiche e aggiornate, migliorando significativamente la qualità delle risposte.

Infine, i servizi RESTful sono particolarmente adatti per automatizzare e scalare l'uso degli LLMs in applicazioni distribuite. Grazie alla loro compatibilità con il protocollo HTTP e la loro facilità di implementazione, i servizi REST possono essere utilizzati per orchestrare l'elaborazione di grandi volumi di richieste simultanee, bilanciando il carico tra diverse istanze dell'LLM e garantendo che le risposte siano fornite in tempi rapidi. Questo è cruciale per applicazioni come i chatbot su larga scala, dove l'interazione in tempo reale con gli utenti è essenziale.

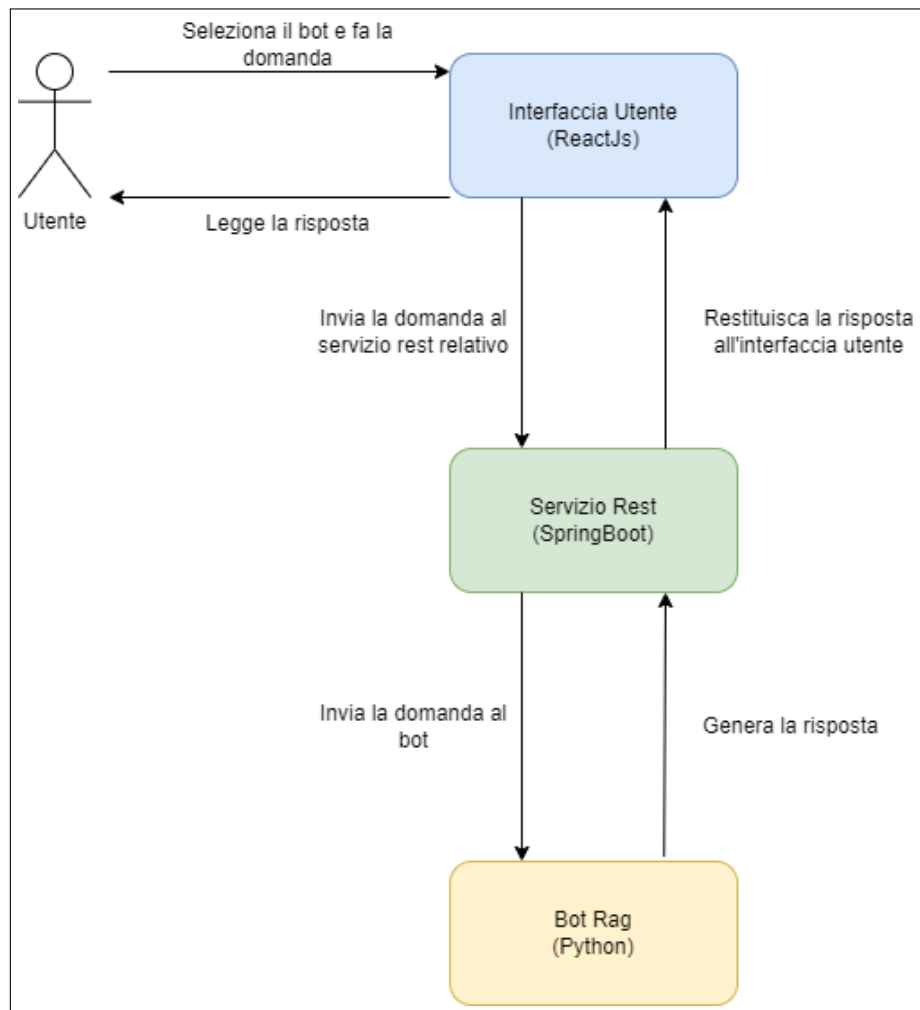
In sintesi, l'integrazione di LLMs con servizi RESTful offre una soluzione potente e flessibile per implementare capacità avanzate di elaborazione del linguaggio naturale in applicazioni moderne. Grazie alla loro modularità, scalabilità e facilità di integrazione, i servizi RESTful non solo facilitano la comunicazione con i modelli di intelligenza artificiale, ma permettono anche di sfruttare appieno le loro potenzialità in un'ampia gamma di scenari applicativi.

Capitolo 2

Analisi e progettazione

Per costruire un sistema robusto e altamente performante, ho integrato tre tecnologie chiave che costituiscono la base della nostra infrastruttura avanzata. L'analisi approfondita delle interazioni tra queste tecnologie ha guidato la progettazione e l'implementazione del sistema.

Questa soluzione consente, tramite un'interfaccia web intuitiva, di selezionare tra diversi bot, ciascuno specializzato in tematiche specifiche di rilevanza per l'azienda. La possibilità di scegliere il bot più adatto alle esigenze dell'utente garantisce risposte precise e contestuali, ottimizzando l'efficienza operativa complessiva. Di seguito viene presentato un esempio di interazione tra l'utente e il sistema.



2.1 Scelta tecnologica

L'infrastruttura del sistema si basa su tre tecnologie principali:

- ReactJs
- SpringBoot
- Python

ReactJS è stato scelto per la creazione dell'interfaccia grafica del sistema. Questa libreria JavaScript, sviluppata da Facebook, è rinomata per la sua capacità di costruire interfacce utente moderne e reattive. Grazie alla sua architettura basata sui componenti e al virtual DOM, ReactJS offre un'esperienza utente fluida e

altamente interattiva, facilitando la gestione e l'aggiornamento dinamico dei dati all'interno dell'applicazione.

Spring Boot è utilizzato per lo sviluppo dei servizi RESTful che costituiscono il cuore del backend del sistema. Questa potente piattaforma Java semplifica la creazione di applicazioni stand-alone e basate su microservizi. Spring Boot consente di costruire API REST robuste e scalabili, gestendo in modo efficiente le comunicazioni tra il frontend e i vari componenti del backend, e garantendo un'architettura modulare e facilmente estendibile.

Python, infine, è impiegato per la creazione e l'addestramento dei bot. Conosciuto per la sua sintassi chiara e la vasta gamma di librerie specializzate, Python è particolarmente adatto per il machine learning e l'elaborazione del linguaggio naturale. Le librerie di Python, come TensorFlow e PyTorch, permettono di sviluppare modelli avanzati per i bot, che possono essere addestrati per rispondere in modo preciso e contestuale alle esigenze degli utenti.

2.1.1 Creazione e addestramento dei chatbot

Nell'infrastruttura creata, ogni volta che è necessario sviluppare e addestrare un nuovo chatbot su documenti PDF specifici di una determinata tematica, viene avviato un progetto in Python che utilizza quattro librerie chiave:

- Flask: Questa libreria consente di creare endpoint REST, facilitando la comunicazione tra il chatbot e l'applicazione backend basata su Spring Boot tramite il protocollo REST.
- LangChain e Chroma: Queste librerie sono responsabili della vettorizzazione dei documenti PDF e del recupero delle informazioni rilevanti in risposta ai prompt inseriti dagli utenti attraverso l'interfaccia grafica.
- Ollama: Permette l'interazione tra il programma e il modello di LLM (Large Language Model) installato localmente, gestendo le richieste e le risposte del chatbot.

Parallelamente, viene avviato un progetto in Spring Boot per gestire la comunicazione tra il chatbot e l'interfaccia grafica tramite il protocollo REST, garantendo un'integrazione fluida e coerente tra il frontend e il backend.

Questa architettura consente di creare chatbot altamente specializzati e funzionali, ottimizzando la gestione delle informazioni e migliorando l'interazione con gli utenti.

Capitolo 3

Titolo capitolo 3

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

3.1 Titolo Paragrafo

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

3.1.1 Titolo Sottoparagrafo

At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga. Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil impedit quo minus id quod maxime placeat facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum hic tenetur a sapiente delectus, ut aut reiciendis voluptatibus maiores alias consequatur aut perferendis doloribus asperiores repellat [?].

Capitolo 4

Titolo capitolo 4

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

4.1 Titolo Paragrafo

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

4.1.1 Titolo Sottoparagrafo

At vero eos et accusamus et iusto odio dignissimos ducimus qui blanditiis praesentium voluptatum deleniti atque corrupti quos dolores et quas molestias excepturi sint occaecati cupiditate non provident, similique sunt in culpa qui officia deserunt mollitia animi, id est laborum et dolorum fuga. Et harum quidem rerum facilis est et expedita distinctio. Nam libero tempore, cum soluta nobis est eligendi optio cumque nihil impedit quo minus id quod maxime placeat facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum hic tenetur a sapiente delectus, ut aut reiciendis voluptatibus maiores alias consequatur aut perferendis doloribus asperiores repellat [?].

Capitolo 5

Conclusioni

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.